WHAT IS CLAIMED IS:

1. An iterative mantissa calculator for a floating point divide and square root processor that selectively calculates a divide result mantissa based on a divisor mantissa and a dividend mantissa or a square-root result mantissa based on an operand mantissa, the calculator comprising:

a plurality of summing devices, each of the summing devices being arranged to generate a partial remainder during a divide operation and at least one of the summing devices being arranged to generate a partial remainder during a square-root operation; and

a selector coupled to each summing device, each selector being arranged to select a quotient bit during a divide operation, and at least one of the selectors being arranged to select a result bit during a square-root operation;

a first of the summing devices being responsive during a first iteration to a partial remainder generated during a prior iteration and to an accumulated quotient generated by a second selector to generate a first partial remainder for a divide operation, a first selector being responsive to the first partial remainder to generate a quotient bit and accumulate a first quotient, a second of the summing devices being responsive during a second

iteration to the first partial remainder generated during a prior iteration and to the accumulated first quotient to generate a second partial remainder for a divide operation and the second selector being responsive to the second partial remainder and the accumulated first quotient to accumulate a second quotient, and

the first of the summing devices being responsive to a partial remainder generated by the second summing device during a prior iteration and to an result accumulated during the prior iteration to generate a sum, the second of the summing devices being responsive to the sum and to a shifted accumulated result to generate a partial remainder and the second selector being responsive to the partial remainder to generate a result bit and accumulate a result.

2. The mantissa calculator of claim 1, further including:

an output responsive to one of the summing devices for outputting the partial remainder generated by the one summing device and the accumulated quotient or result.

3. The mantissa calculator of claim 1, including a result adder responsive to the partial remainder and accumulated quotient or result to

generate a quotient mantissa or square-root result mantissa.

4. The mantissa calculator of claim 1, wherein there are two summing devices, and during each iteration in the divide mode the respective first and second summing devices each calculates a respective partial remainder, W[j+1], for use by the other of the first and second summing devices during the next iteration, represented by $2*W[j]-S_{j+1}*D$, where W[j] is the partial remainder generated by one summing device during the current iteration, S_{j+1} is a result bit based on the partial remainder generated by the other summing device during a prior iteration and D is a divisor bit, and wherein the partial remainder, W(0), for an initial iteration is equal to X-D, where X is the respective dividend bit.

5. The mantissa calculator of claim 4, further including:

an output responsive to the second summing device for outputting the partial remainder generated by the second summing device and the accumulated quotient.

6. The mantissa calculator of claim 4, wherein during the square root mode, the first summing device calculates a sum representative of $2W[j]-2S[j]S_{j+1}$, and the second summing device calculates a partial

remainder W[j+1] based on the sum and $S_{j+1}^{2} \bullet 2^{-(j+1)}$, where W[j] is the partial remainder calculated during a prior iteration, S[j] is the accumulated result bits to the current iteration, and S_{j+1} is the result bit based on a partial remainder calculated by the second summing device during the prior iteration.

7. The mantissa calculator of claim 6, further including:

an output responsive to the second summing device for outputting the partial remainder generated by the second summing device and the accumulated result.

8. The mantissa calculator of claim 6, including a shifter responsive to plural result bits based on the partial remainder of the second summing device for calculating $S_{j+1}^{2} \bullet 2^{-(j+1)}$.

9. The mantissa calculator of claim 1, wherein during the square root mode, the first summing device calculates a sum representative of $2W[j]-2S[j]S_{j+1}$, and the second summing device calculates a partial remainder W[j+1] based on the sum and $S_{j+1}^{2} \bullet 2^{-(j+1)}$, where W[j] is the partial remainder calculated during a prior iteration, S[j] is the accumulated result bits to the current iteration, and S_{j+1} is the result bit

based on a partial remainder calculated by the second summing device during the prior iteration.

10. The mantissa calculator of claim 9, including a shifter responsive to plural result bits based on the partial remainder of the second summing device for calculating $S_{j+1}^2 \bullet 2^{-(j+1)}$.

11. A computer processor for calculating a floating point quotient based on a divisor mantissa, a dividend mantissa, a divisor exponent and a dividend exponent, and for calculating a square root result based on an operand mantissa and an operand exponent, the processor comprising:

an iterative mantissa calculator that selectively calculates a divide result mantissa or a square-root result mantissa, the calculator having:

a plurality of summing devices, each of the summing devices being arranged to generate a partial remainder during a divide operation and at least one of the summing devices being arranged to generate a partial remainder during a square-root operation; and

a selector coupled to each summing device, each selector being arranged to select a quotient bit during a divide operation, and at least one of the selectors being arranged to select a quotient bit during a divide operation, and at least one of the selectors being arranged

to select a result bit during a square-root operation;

a first of the summing devices being responsive during a first iteration to a partial remainder generated during a prior iteration and to an accumulated quotient generated by a second selector to generate a first partial remainder for a divide operation, a first selector being responsive to the first partial remainder to generate a quotient bit and accumulate a first quotient, a second of the summing devices being responsive during a second iteration to the first partial remainder generated during a prior iteration and to the accumulated first quotient to generate a second partial remainder for a divide operation and the second selector being responsive to the second partial remainder and the accumulated first quotient to accumulate a second quotient, and

the first of the summing devices being responsive to a partial remainder generated by the second summing device during a prior iteration and to an result accumulated during the prior iteration to generate a sum, the second of the summing devices being responsive to the sum and to a shifted accumulated result to generate a partial remainder and the second selector being responsive to the partial

remainder to generate a result bit and accumulate a result; and

an exponent calculator responsive to the divisor and dividend exponents to calculate a divide exponent and being responsive to the operand exponent to calculate the square-root exponent.

12. The processor of claim 11, including a result adder responsive to the partial remainder and accumulated quotient or result to generate a quotient mantissa or square-root result mantissa.

13. The processor of claim 11, wherein there are two summing devices, and during each iteration in the divide mode the respective first and second summing devices each calculates a respective partial remainder, W[j+1], for use by the other of the first and second summing devices during the next iteration, represented by $2*W[j]-S_{j+1}*D$, where W[j] is the partial remainder generated by one summing device during the current iteration, S_{j+1} is a result bit based on the partial remainder generated by the other summing device during a prior iteration and D is a respective divisor bit, and wherein the partial remainder, W(0), for an initial iteration is equal to X-D, where X is the respective dividend bit.

14. The processor of claim 13, further including:

an output responsive to the second summing device for outputting the partial remainder generated by the second summing device and the accumulated quotient.

15. The processor of claim 13, wherein during the square root mode, the first summing device calculates a sum representative of $2W[j]-2S[j]S_{j+1}$, and during a second iteration of the square root mode, the second summing device calculates a second partial remainder $W[j+1]$ based on the first partial remainder and $S_{j+1}^2 \bullet 2^{-(j+1)}$, where $W[j]$ is the partial remainder calculated during a prior iteration, $S[j]$ is the accumulated result bits to the current iteration, and S_{j+1} is the result bit based on a partial remainder calculated by the second summing device during the prior iteration.

16. The processor of claim 15, further including:

a output responsive to the second summing device for outputting the partial remainder generated by the second summing device and the accumulated result.

17. The processor of claim 15, including a shifter responsive to plural result bits based on the partial remainder of the second summing device for calculating $S_{j+1}^2 \bullet 2^{-(j+1)}$.

18. A computer process of selectively performing a floating point divide or square root operation, including steps of:

a) providing an iterative mantissa calculator for operating in a divide mode and in a square root mode, the calculator having at least first and second summing devices;

b) operating the calculator in the divide mode, including steps of:

b1) operating the first summing device during a first iteration to generate a first divide partial remainder based on a divide partial remainder generated by the second summing device during a prior iteration and an accumulated quotient generated during the prior iteration,

b2) selecting a first quotient bit based on the first divide partial remainder,

b3) generating a first quotient based on the accumulated quotient generated during the prior iteration and the first quotient bit,

b4) operating the second summing device during a second iteration to generate a second divide partial remainder

based on the first divide partial remainder and the first quotient,

b5) selecting a second quotient bit based on the second divide partial remainder, and

b6) generating a second quotient based on the accumulated first quotient and the second quotient bit; and

c) operating the calculator in the square root mode, including steps of:

c1) operating the first summing device to generate a sum based on a square-root partial remainder generated by the second summing device during a prior iteration and an accumulated result generated during a prior iteration;

c2) operating the second summing device during a second iteration to generate a square-root partial remainder based on the sum, the accumulated result generated during prior iterations, and an operand mantissa,

c3) selecting a result bit based on the square-root partial remainder, and

c4) generating a result based on the accumulated result and the result bit.

19. The process of claim 18, wherein step (b) further includes steps of:

b5) output the second divide partial remainder, and

b6) output the accumulated quotient,

and step (c) further includes:

c4) output the second square-root partial remainder, and

c5) output the accumulated result.

20. The process of claim 18, wherein step (b1) calculates the first divide partial remainder, W[j+1], as 2*W[j]-S_{j+1}*D, wherein step (b3) calculates the second divide partial remainder, W[j+2], as 2*W[j+1]-S_{j+2}*D, wherein step (c1) calculates the sum as 2W[j]-2S[j]S_{j+1}, and wherein step (c2) calculates the square-root partial remainder W[j+1] as $2W[j]-2S[j]S_{j+1}-S_{j+1}^{2}\bullet 2^{-(j+1)}$, where W[j] is the partial remainder generated by the second summing device during the prior iteration, S_{j+1} is a second quotient bit or result bit generated by the second summing device during a prior iteration, S_{j+2} is a second quotient bit generated by the first summing device during the first iteration, S[j] is the accumulated result, and D is a divisor.